```python

From django.http import HttpResponse

From django.shortcuts import render

Def image\_response\_view(request):

# Replace ‘path\_to\_your\_image.jpg’ with the actual path to your image file

With open(‘path\_to\_your\_image.jpg’, ‘rb’) as f:

Image\_data = f.read()

Return HttpResponse(image\_data, content\_type=’image/jpeg’)

```

In this example:

- Replace `’path\_to\_your\_image.jpg’` with the path to your actual image file. Make sure the path is correct and the image file exists in your Django project directory or accessible path.

- `open(‘path\_to\_your\_image.jpg’, ‘rb’)` opens the image file in binary mode (‘rb’ stands for read binary).

- `image\_data = f.read()` reads the image file into `image\_data`.

- `HttpResponse(image\_data, content\_type=’image/jpeg’)` creates an HTTP response object with `image\_data` as the content and sets the content type to `image/jpeg`. Adjust `image/jpeg` to `image/png` or `image/gif` depending on the type of image you are serving.

Remember to define a URL pattern in your Django `urls.py` to map to this view, so that when a request is made to that URL, the image will be served as the response.

For example, in your `urls.py`:

```python

From django.urls import path

From . import views

Urlpatterns = [

Path(‘image/’, views.image\_response\_view, name=’image\_response’),

]

```

This will map the URL `/image/` to the `image\_response\_view` function defined earlier. When you visit `/image/` in your browser, Django will return the image specified by `’path\_to\_your\_image.jpg’`.